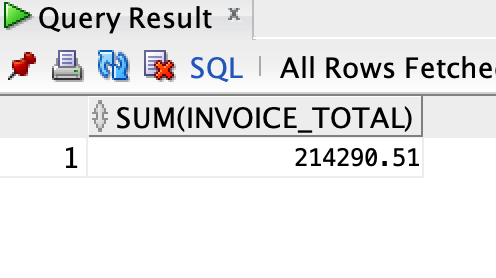
Assignment 2

Due Date Oct 8th

Please use the data provided below for solving the following questions.

1. (2 points) Without the use of ‘GROUP BY’ write a query to obtain the following:
   1. Sum of invoice\_total

SELECT SUM(INVOICE\_TOTAL) FROM INVOICES



* 1. Average invoice\_total to three decimal digits only

SELECT ROUND(AVG(INVOICE\_TOTAL),3) FROM INVOICES

A screenshot of a computer

Description automatically generated

* 1. Count of distinct vendors

SELECT COUNT(DISTINCT VENDOR\_ID) FROM INVOICES

A screenshot of a computer

Description automatically generated

* 1. Standard Deviation of invoice\_total

SELECT ROUND(STDDEV(INVOICE\_TOTAL),3) FROM INVOICES

A screenshot of a computer

Description automatically generated

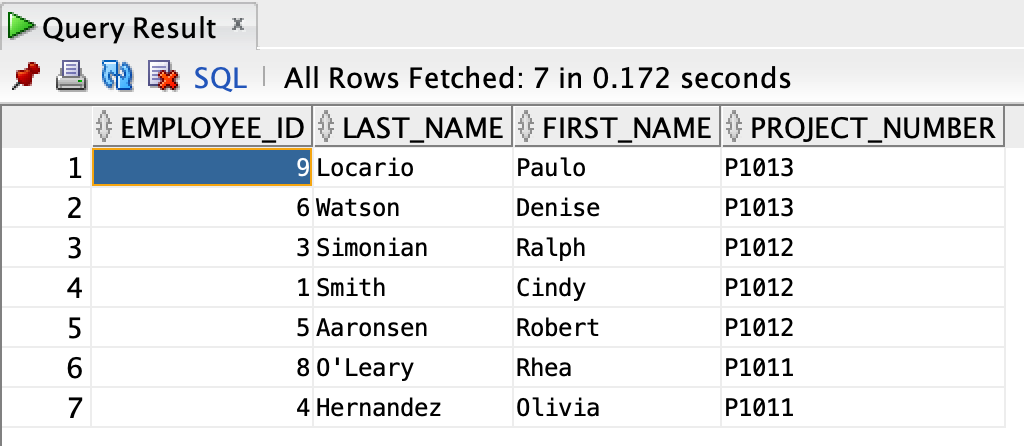
1. (2 points) Write a query that joins *Employees* and *Projects* tables and returns all matching rows (i.e. only employees that have projects assigned and only projects that have employees assigned) and use table alias in the query.
   1. Update the select to only return the employee\_id, first and last name, and the project number.
   2. Sort the data by project number so we group employees by project.

SELECT e.EMPLOYEE\_ID, e.LAST\_NAME, e.FIRST\_NAME, j.PROJECT\_NUMBER

FROM EMPLOYEES e

JOIN PROJECTS j on e.EMPLOYEE\_ID = j.EMPLOYEE\_ID

ORDER BY j.PROJECT\_NUMBER DESC;



1. (2 points) Write a query that returns vendor\_id, invoice\_number, invoice\_due\_date, payment\_date from the invoices table but only for invoices that have a payment date.

SELECT VENDOR\_ID, INVOICE\_NUMBER,INVOICE\_DUE\_DATE,PAYMENT\_DATE

FROM INVOICES

WHERE PAYMENT\_DATE IS NOT NULL;

A screenshot of a computer

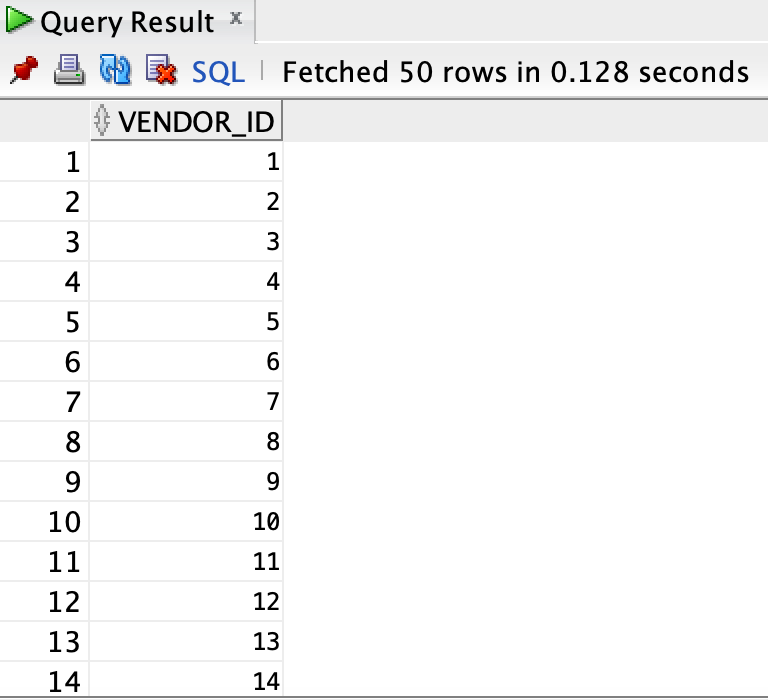
Description automatically generated

1. (2 points) Write a query that generates a single column with a distinct list of vendor\_ids from the *Vendors* table. Similarly write a query that pulls the distinct list of vendors from the *Invoices* table.
   1. Using the command ‘MINUS’ command, find the difference between the two.
   2. Explain the result of the MINUS

SELECT DISTINCT VENDOR\_ID FROM VENDORS

MINUS

SELECT DISTINCT VENDOR\_ID FROM INVOICES;



The usage of MINUS in this query will result in a list of VENDOR\_ID values from the VENDORS table that do not exist in the INVOICES table.

1. (2 points) First find the average list price for all the products on the *Products* table. Using this query as a subquery, find all order\_items records where the item\_price is greater that the average calculated in the previous query.

SELECT \* FROM ORDER\_ITEMS

WHERE ITEM\_PRICE > (SELECT AVG(LIST\_PRICE) FROM PRODUCTS)

A screenshot of a computer

Description automatically generated